different versions of an application by using the conversion stack according to another embodiment of the present invention.

[0094] As shown, in step **1001**, a document of a lower version from the application of the lower version is received.

[0095] In step **1002**, the document of the lower version is successively converted into zero, one or more documents of intermediate versions between the higher version and the lower version and a document of the higher version by using one or more differentiation models between data models between the higher version and the lower version in the conversion stack, and the document of the lower version and the zero, one or more documents of the intermediate versions between the higher version and the lower version or the relevant information therein are stored.

[0096] In step **1003**, the document of the higher version is provided to the application of the higher version so that required modifications can be made to the document of the higher version by the application of the higher version.

[0097] In step **1004**, the modified document of the higher version is successively converted into zero, one or more documents of the intermediate versions and a modified document of the lower version by using one or more differentiation models between data models between the higher version and the lower version in the conversion stack, and by merging successively with the zero, one or more stored documents of the intermediate versions between the higher version and the lower version and the stored document of the lower version or the relevant information therein.

[0098] In step **1005**, the modified document of the lower version is sent to the application of the lower version.

[0099] According to still another embodiment of the present invention, the step **802** for converting different versions of documents of the application by the conversion stack includes the following sub-steps: first, obtaining a document of a first version; second, successively converting the document of the first version into zero, one or more documents of intermediate versions between the first version and the second version and a document of the second version by using one or more differentiation models between data models between the first version and the second version in the conversion stack; finally, providing the document of the second version.

[0100] According to an embodiment of the present invention, the conversion further includes verifying a converted document according to the data model of the version as the conversion target, so as to make the converted document conform to the data model.

[0101] The above described a method for providing inter-version document compatibility according to an embodiment of the present invention. It should be pointed out that, the above description is only exemplarily, rather than limitation to the present invention. Compared with that is described, the method can have more, fewer or different steps, and the orders between the steps can be different or they can be executed in parallel. For example, in some embodiments of the present invention, the method may exclude one or more of the above optional steps.

[0102] The present invention can be realized by hardware, software, or a combination thereof. The present invention can be implemented in a single computer system in a centralized manner, or in a distributed manner in which different components are distributed in several inter-connected computer systems. Any computer system or other devices suitable for executing the method described herein are all suitable. A typical combination of hardware and software can be a general-purpose computer system with a computer program, which when being loaded and executed, controls the computer system to execute the method of the present invention and constitutes the apparatus of the present invention.

[0103] The present invention can also be embodied in a computer program product, which includes all the features that enable to realize the methods described herein, and when being loaded into a computer system, can execute the method.

[0104] Although the present invention has been illustrated and described with reference to preferred embodiments, those skilled in the art will appreciate that various changes in form and details can be made thereto without departing from the spirit and scope of the present invention.

What is claimed is:

1. A method performed in a data processing machine for providing inter-version document compatibility, comprising the steps of:

providing a conversion stack which includes differentiation models between data models of different versions of an application; and

converting a document of one version of an application to a document of another version by using the conversion stack to provide compatibility between the documents of the different versions of the application.

2. The method of claim **1**, wherein the differentiation models between data models of different versions of the application are differentiation models between data models of adjacent versions of the application.

3. The method of claim **1**, wherein the converting step further comprises:

obtaining a document of a first version;

successively converting the document of the first version into at least one document of an intermediate version between the first version and a second version and a document of the second version by using one or more differentiation models between data models between the first version and the second version in the conversion stack; and

providing the document of the second version.

4. The method according to claim **1**, wherein the converting step further comprises:

obtaining a document of a higher version;

successively converting the document of the higher version into documents of intermediate versions between the higher version and a lower version and a document of the lower version by using at least one differentiation model between data models between the higher version and the lower version in the conversion stack, and storing the document of the higher version and the documents of intermediate versions between the higher version and the lower version or the relevant information therein;

sending the document of the lower version to an application of the lower version so that required modifications to the document of the lower version may be performed by the application of the lower version;

receiving a modified document of the lower version from the application of the lower version; and

successively converting the modified document of the lower version into modified documents of intermediate versions and a modified document of the higher version by using at least one differentiation model between data models between the higher version and the lower version in the conversion stack, and by successively merging